

UNITED STATES PATENT AND TRADEMARK OFFICE  
DOCUMENT CLASSIFICATION BARCODE SHEET

---



000

CATEGORY:

**CLEARED**

---

ADDRESS  
CONTACT IF FOUND:

## TRANSMITTAL FORM FOR FILING PATENT APPLICATION



Attorney  
Docket No.: ADI-005XX

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP  
Ten Post Office Square  
Boston, Massachusetts 02109  
Telephone: (617) 542-2290  
Telecopier: (617) 451-0313



Express Mail No: EL418425050US

BOX PATENT APPLICATION  
Assistant Commissioner for Patents  
Washington, D.C. 20231

Date: August 1, 2000

First Named Inventor or Application  
Identifier: Marc Hoffman et al.

Sir:

Transmitted herewith under 37 CFR § 1.53 for filing is the patent application of:

Inventor: Marc Hoffman and Jose Fridman

Entitled: A METHOD FOR EFFICIENTLY COMPUTING A FAST FOURIER TRANSFORM

[ ] This is a request for filing a [ ] **continuation** [ ] **divisional** [ ] **continuation**  
**in-part** application under §1.53(b) of prior Application No. \_\_\_\_\_, filed  
\_\_\_\_\_ entitled:

Enclosed are:

[X] 18 pages of written description, claims and Abstract, inclusive

[X] 5 sheets of [ ] informal [x] formal drawings of Figs. 1-7 (one set)

[X] Oath or Declaration

[x] Newly executed (original or copy)

[ ] Copy from prior application (37 CFR 1.63(d)) (for continuation/divisional).

The entire disclosure of the prior application, from which a copy of the oath  
or declaration is supplied, is considered as being part of the disclosure of  
the accompanying application and is hereby incorporated by reference therein.

[ ] Declaration to be filed later

[X] Cover sheet and Assignment of the invention to: ANALOG DEVICES, INC.

[ ] Certified copy of a \_\_\_\_\_ application (if foreign priority is  
claimed) with letter claiming priority under Rule 55.

[ ] Information Disclosure Statement with \_\_\_ citations

[ ] Preliminary amendment is enclosed.

[X] Return receipt postcard

[ ] Other:

**TRANSMITTAL FORM FOR FILING PATENT APPLICATION (CONTINUED)**

Attorney

Docket No.: ADI-005XX

- ☐ Verified statement of Small Entity status (§1.9 and §1.27)
- ☐ Verified statement of Small Entity was filed in prior application. Status still proper and desired
- ☐ Priority is claimed under 35 USC § 120 as indicated on the attached sheet 4.
- ☐ Priority is claimed under 35 USC §119(a)-(d) as indicated on the attached sheet 4.
- ☐ Priority is claimed under 35 USC §119 (e) as indicated on the attached sheet 4.
- ☐ \_\_\_\_\_ is hereby appointed Associate Attorney by:  
Registration No.:

\_\_\_\_\_  
Attorney of Record  
Registration No.:

- ☐ **Power of Attorney** in the originally-filed application has been granted to one or more of the registered attorneys listed below. The attorneys listed below not previously granted power in the originally-filed application, as well as \_\_\_\_\_, are hereby given associate power:

Registration No.:

Stanley M. Schurgin, Reg. No. 20,979  
Charles L. Gagnebin III, Reg. No. 25,467  
Paul J. Hayes, Reg. No. 28,307  
Victor B. Lebovici, Reg. No. 30,864

Eugene A. Feher, Reg. No. 33,171  
Beverly E. Hjorth, Reg. No. 32,033  
Holliday C. Heine, Reg. No. 34,346  
Gordon R. Moriarty, Reg. No. 38,973

- ☐ Cancel in this application original claims \_\_\_\_\_ of the prior application before calculating the filing fee.
- ☐ Add in this application claims \_\_\_\_\_ per amendment before calculating fee.

CLAIMS FILED:	MINUS BASE:	EXTRA CLAIMS:	RATE:	BASIC FEE:
				\$690.00
Independent	3 - 3	= 0	x \$78.00 =	0.00
Total	8 - 20	= 0	x \$18.00 =	0.00
<input type="checkbox"/> Multiple Dependent Claims (1st presentation)			+ \$260.00 =	0.00
SUBTOTAL FILING FEE				\$690.00
Small Entity filing, divide by 2. (Note: verified statement must be attached per §1.9, §1.27, §1.28.)				0.00
TOTAL Filing Fee				\$690.00

Attorney Docket No.: ADI-005XX


**TRANSMITTAL FOR FILING PATENT APPLICATION (CONTINUED)**

- [X] The filing fee has been calculated above; a check in the amount of \$690.00 is enclosed.
- [ ] The filing fee will be submitted at a later date.
- [X] In the event a Petition for Extension of Time under 37 CFR §1.17 is required by this paper and not otherwise provided, such Petition is hereby made and authorization is provided herewith to charge Deposit Account No. 23-0804 for the cost of such extension.
- [X] The Commissioner is hereby authorized to charge payment of any additional filing fees under 37 CFR §1.16 associated with this communication or credit any overpayment to Deposit Account No. 23-0804.

**[X] Customer Number 207**

Address all future communications to:

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP  
Ten Post Office Square  
Boston, Massachusetts 02109  
Telephone: (617) 542-2290  
Telecopier: (617) 451-0313

  
\_\_\_\_\_  
Attorney of Record: Beverly E. Hjorth  
Registration No. 32,033

Attorney Docket No.: ADI-005XX

**TRANSMITTAL FOR FILING PATENT APPLICATION (CONTINUED)**

☐ Priority is claimed under 35 USC § 120 of prior Application(s)  
No. \_\_\_\_\_, filed \_\_\_\_\_, entitled:

☐ The above-identified application(s) is/are assigned of record to:

☐ Priority is claimed under 35 USC § 119 (a)-(d) of the following application(s).

\_\_\_\_\_  
(Application Number) (Country) (Filing Date)

\_\_\_\_\_  
(Application Number) (Country) (Filing Date)

\_\_\_\_\_  
(Application Number) (Country) (Filing Date)

☐ The above-identified application(s) is/are assigned of record to:

☐ Priority is claimed under 35 USC § 119 (e) of the following provisional application(s).

\_\_\_\_\_  
(Application Number) (Filing Date)

\_\_\_\_\_  
(Application Number) (Filing Date)

\_\_\_\_\_  
(Application Number) (Filing Date)

☐ The above-identified provisional application(s) is/are assigned of record to:

☐ The claim of small entity status in the above-identified provisional application(s) is made in this application and a copy of the small entity form(s) from the provisional application(s) is/are enclosed.

BEH/rec

Enc.

SUBMIT IN TRIPLICATE

228022

TITLE OF THE INVENTION

A METHOD FOR EFFICIENTLY COMPUTING A FAST FOURIER TRANSFORM

5

CROSS REFERENCE TO RELATED APPLICATIONS

N/A

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR  
DEVELOPMENT

N/A

10

BACKGROUND OF THE INVENTION

This invention relates generally to a technique for computing a Fast Fourier Transform (FFT) and more particularly to methods and apparatus for computing an FFT in which the number of loop operations are reduced and the resultant output data values from each stage data are stored in a memory with a unity stride.

The fast Fourier transform (FFT) is the generic name for a class of computationally efficient algorithms that implement the discrete Fourier transforms (DFT), and are widely used in the field of digital signal processing.

A band-limited time-varying analog signal can be converted into a series of discrete digital signals by sampling the analog signal at or above the Nyquist frequency, to avoid aliasing, and digitizing the sampled analog signals. A DFT algorithm may be applied to these digitized samples to calculate the discrete frequency components contained within the analog signal. The DFT algorithm provides, as output data values, the magnitude and phase of the discrete frequency components of the analog signal. These discrete frequency components are evenly spaced between 0 and  $\frac{1}{2}$  the sampling frequency, which is

typically the Nyquist sampling frequency. The number of discrete frequency components is equal to the number of the digitized samples that are used as input data. For example, a DFT having 8 input samples, will have 8 evenly spaced frequency components as output.

The DFT is given by:

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{j \frac{2\pi nk}{N}}$$

where:

N is the number of input samples;

n is the particular index in the time domain sample from n=0 to n=N-1;

x(n) is the magnitude of the time domain analog signal at the time sample point corresponding to n;

k is the particular frequency domain component from k=0 to k=N-1; and

X(k) is the magnitude of the frequency component corresponding to the frequency index k.

The DFT involves a large number of calculations and memory operations and, as such, is not computationally efficient. The FFT algorithm reduces the computational load of calculating the discrete frequency components in a time domain signal from approximately  $6(N^2)$  to approximately  $N \log_2 N$ . As will be discussed in detail below, this reduction in the number of calculations is achieved by decomposing the standard DFT algorithm into a series of smaller and smaller DFTs. For example, an 8 point DFT can be decomposed into an FFT involving 3 stages of calculations. In this manner the 8 point FFT is decomposed into one 8 point FFT that can be decomposed into two 4 point DFTs that are decomposed into four 2 point DFTs.

At each stage of the FFT algorithm the canonical mathematical operations performed on each pair of input data is known as the FFT butterfly operation. Figure 4 illustrates the canonical FFT butterfly operations which are

5 
$$\begin{aligned} X(m+1) &= X(m) + W(n,k)Y(m) \\ Y(m+1) &= X(m) - W(n,k)Y(m) \end{aligned}$$

where X and Y are input signals and are discussed in more detail below.  $W(n,k)$  (the "twiddle factor") is a complex value and is given by the formula:

$$W(n,k) = e^{j\frac{2\pi nk}{N}}.$$

10 This complex function is periodic and for an FFT of a given size N, provides N/2 constant values. As discussed in more detail below, these values may be pre-calculated and stored in a memory.

Figure 1 illustrates a traditional decimation in time  
15 FFT signal flow graph for an 8 input (8 point) FFT. An FFT algorithm will include  $\log_2 N$  stages of calculations. Thus, the 8 point FFT signal flow graph 100 is divided into  $\log_2 8$ , or three, stages: the first stage 102, second stage 104 and the third stage 106, where each stage performs N/2 butterfly  
20 calculations. Thus, in Fig. 1, every stage of the signal flow graph 100 will calculate 8/2, or 4 butterfly calculations per stage. An examination of Fig.1 also shows that the first stage provides four 2 point FFTs, the second stage provides two 4-point FFT's and the final stage  
25 provides one 8-point FFT. Thus, each stage will have a number of groups in which the FFTs are calculated. The number of groups per stage is given by:

$$\text{groups} = 2^{\log_2(N) - m}$$

where N is the number of input data points, and m is  
30 the number of the stage and is from  $m=1$  to  $m=\log_2 N$ . Thus in



Fig. 1, the first stage has  $2^{3-1}$  or 4 groups of FFTs, 108, 110, 112, and 114. The second stage has  $2^{3-2}$  or 2 groups of FFTs, 116, and 118. The final stage has  $2^{3-3}$  or 1 group of an FFT 120.

5 To compute an FFT on a computer, the signal flow graph 100 must be translated into a software program. A software program based on the traditional FFT signal flow graph will first typically re-order the data into a bit-reversed order as shown by the input data 122. Next, three loops that  
10 calculate FFT data are executed. The outermost loop, known as the stage loop, will be executed only for each stage. Therefore, for an N point FFT, there will be  $\log_2 N$  outer loops that must be executed. The middle loop, known as the group loop, will be executed a different number of times for  
15 each stage. As discussed above, the number of groups per stage will vary from  $2^{\log_2(N)-m}$  to 1 depending on the position of the stage in the algorithm. Thus for the early stages of the FFT the group loop will be entered into and out of many times in each stage. The inner most loop, known as the  
20 butterfly loop, will be executed  $N/2$  times for each stage.

The FFT signal flow graph 100 also illustrates another aspect of the traditional FFT technique. The data that is provided by each butterfly calculator is stored in a different sequence in each stage of the FFT. For example,  
25 in the first stage 102 the input data is stored in a bit-reversed order. Thus, each butterfly calculator receives input data values that are stored in adjacent memory locations. In addition, each butterfly calculator provides output data values that are stored in adjacent memory  
30 locations in the sequence in which they are calculated. In the second stage 104 each butterfly calculation receives

input data that is separated by 2 storage locations, and the output data values are stored in memory locations that are also 2 storage locations apart. In the third stage 106, each butterfly calculation receives data that is 4 storage locations apart and provides output data values that are also stored 4 storage locations apart. Thus, the distance between the storage locations where the output data values are stored (the stride) varies as a power of two from  $2^0$  to  $2^{N/2}$ . Thus, in the illustrative embodiment the stride varies between 1 and 4 as discussed above.

In a typical computing system, the most time consuming operations are the reading and writing of data to and from memory respectively. Since the FFT is a very data intensive algorithm, many schemes have been developed to optimize the memory-addressing problem. Typically memory systems have been designed to increase the performance of the FFT by changing the pattern of how the memory is stored, by using smaller faster memories for the data, or by dedicating specific hardware to calculate the desired memory locations. However, the very nature of the traditional FFT as shown in Fig. 1 illustrates the limitations of these approaches. For each stage, a new stride will have to be computed and, for each stage, there are only so many ways to change the pattern of the memory storage. Modern computer languages also allow the accessing of memory locations directly using "pointers". Pointer arithmetic can be time consuming as well and the need to recalculate the pointer arithmetic for each stage is inefficient.

In addition to the data storage problem, traditionally, the control and overhead processing for a computer program takes up the bulk of the program memory, but only a small

fraction of the actual processing time. Therefore, minimizing the control and overhead portions of a computer program is one method to further optimize the memory usage of the program. As discussed above, in the signal flow graph 100 the number of the stage loops and the butterfly loops to be executed are set by the system parameters, in particular the number of input data points used. The number of group loops to be executed however changes with each stage. In particular, in the early stages of the algorithm, the overhead and control software will be executing a large number of group loops each having a small number of butterfly loops for each stage. This entering and exiting of the group loops will result in a complex iteration space in which a large number of overhead and control instructions need to be executed, resulting in an inefficient program execution.

It would therefore be desirable to be able to compute an FFT in a manner that reduces the number of required iterations and simplifies the calculation of the storage locations of the output data values from each stage in memory.

#### BRIEF SUMMARY OF THE INVENTION

The present invention provides a method and apparatus for computing a FFT in which a unity stride is used to store the output data from each stage, and each stage of the FFT does not include a group loop calculation stage.

A method for computing an FFT is disclosed in which a sequence of first data points is received and stored in a first memory area. An FFT butterfly calculator selects R input data from the sequence of first data points where the

input data are separated by N/R data points. The FFT butterfly calculator also receives the appropriate twiddle factors that are stored in sequential locations in a bit reversed order in a second memory area. The FFT butterfly calculator calculates a radix R butterfly calculation and stores the output data values in a third memory in the sequence in which they are calculated.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING

The invention will be more fully understood from the following detailed description taken in conjunction with the accompanying drawings in which:

Fig. 1 is a schematic illustration of a signal flow diagram of a traditional FFT;

Fig. 2 is a block diagram illustrating an illustrative embodiment of the present invention;

Fig. 3 is a signal flow diagram illustrating the signal flow diagram of an exemplary FFT calculation stage of the present invention;

Fig. 4a is a signal flow diagram illustrating an FFT butterfly calculator;

Fig. 5 is a graphical illustration of the twiddle factors for an 8 point FFT;

Fig. 6 is a signal flow diagram for an 8 point FFT consistent with the present invention; and

Fig. 7 is a table illustrating the storing of twiddle factors in bit reversed order.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

[illegible]

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

Fig. 3 illustrates one embodiment of a signal flow diagram of an FFT calculator stage 300 consistent with the presently disclosed FFT method. The signal flow diagram illustrating the FFT calculator stage 300 will be identical for any stage of the presently disclosed FFT method. Thus, every stage of the FFT method consistent with the presently disclosed FFT method will have a signal flow diagram geometry that is the same as every other stage.

In the signal flow graph, input data values 302 are provided to the FFT calculator stage 300. The input data 302 is stored sequentially in the time-order in which each sample is taken. Thus, the input data values 302 represent the sampled and digitized band-limited analog signal in the sequential order in which they were sampled. This is in contrast to the bit-reversed storage required for the traditional FFT algorithm illustrated in Fig. 1. Storing the data sequentially results in a more efficient FFT algorithm because simpler calculations for the memory operations are required for each stage. As discussed above, the fewer memory operations a computer program executes, the more efficient it will be.

The number of input and output data values the FFT butterfly calculator has determines the radix of an FFT. Therefore for the embodiment illustrated in Fig. 3, each FFT butterfly calculator in the FFT calculator stage 300 has two inputs and is a radix 2 FFT. An FFT method that has 4 inputs for each FFT butterfly calculator is a radix 4 FFT. Although the illustrative embodiment described herein is for a radix 2 FFT, it would be obvious to modify the presently disclosed FFT method for an FFT having radices other than 2 and for a FFT having a mixed-radix as well.

Each of the FFT butterfly calculators in the FFT calculator stage 300 shown in Fig 3 requires, as input, two data input values and one twiddle factor value and provides, as output, two output data values. In each stage of the presently disclosed FFT method the input data values for any FFT butterfly calculator are  $N/R$  data points apart. Where  $N$  is the number of input data points and  $R$  is the radix of the FFT. In the embodiment illustrated in Fig. 3, which is an 8-point radix-2 FFT, the two input data values will be  $N/2$ , or 4, data points apart.

Output stage 308 illustrates the storage of the output data values in the order in which it is calculated. The first butterfly calculator uses the input data  $x(0)$  and  $x(4)$ . The output data values from this FFT butterfly are stored in the first and second memory locations 310, 312. The second butterfly calculator uses the input data  $x(1)$  and  $x(5)$ . The output data values from this FFT butterfly are stored in the third and fourth memory locations 314, 316. The third butterfly calculator uses the input data  $x(2)$  and  $x(6)$ . The output data values from this FFT butterfly are stored in the fifth and sixth memory locations 318, 320. The fourth butterfly calculator uses the input data  $x(3)$  and  $x(7)$ . The output data values from this FFT butterfly are stored in the seventh and eighth memory locations 322, 324. Thus, the output data values are "re-ordered" according to the calculation order, unlike the traditional FFT algorithm illustrated in Fig.1.

The re-ordering of the output data values, such that they are written into memory with a unity stride, helps to increase the efficiency of the presently disclosed FFT method. The unity stride reduces the number of calculations

needed for the memory operations and, in addition, simplifies the arithmetic when using pointers or other memory accessing functions.

Fig. 4 illustrates a suitable radix 2 FFT butterfly calculator 400 consistent with the presently disclosed FFT method. Input data Y(m) 402 and X(m) 404 are mathematically manipulated and combined to provide output data values 406 and 408. Input data Y(k) 404 is multiplied, by multiplier 409, with  $W_N^{nk}$  410.  $W_N^{nk}$  is one of a predetermined number of "twiddle factor" constants that are used throughout the FFT method. Thus, for each stage in an FFT, there will be  $N/2$  twiddle factors required. However, as will be explained below, there will not in general be  $N$  unique twiddle factors per stage.

In general, the twiddle factors are complex numbers having real and imaginary parts. The resultant product, which in general is also complex, is added, by adder 412, to input data X(m) 402 to form the output data value 406, and subtracted, by adder 414, from X(m) to form the output data value 408. Thus, in general, the arithmetic of the butterfly calculator is complex and requires complex additions and multiplications.

The twiddle factor data is required for every stage of the FFT method. As shown in figure 5, the twiddle factor values can be illustrated as points that are equidistantly and symmetrically spaced apart on the circumference of the unit circle in the imaginary plane. Thus, each twiddle factor represents a magnitude and a phase since it has both real and imaginary parts. In the presently disclosed FFT method, the number of twiddle factor values is equal to one-half the number of input data values for each FFT butterfly



calculator. Thus, Fig. 5 represents the twiddle factors for an 8 point FFT. However, as is illustrated in Fig. 5, the twiddle factors are symmetric with respect to the origin. Thus, each twiddle factor will be the negative of the real and imaginary parts of another. In the illustrative embodiment,  $W_9^0 = -W_8^4$ ;  $W_9^1 = -W_8^5$ ;  $W_9^2 = -W_8^6$ ; and  $W_9^3 = -W_8^7$ . Since each twiddle factor can be used for either of two values, the memory needed to store the twiddle factor values is decreased by a factor of 2.

Figure 6 illustrates an 8-point radix-2 FFT signal flow diagram 600 according to the presently disclosed FFT method. The FFT signal flow diagram 600 includes 3 stages, 602, 604, 606, and each stage has four FFT butterfly calculators per stage. Thus, each stage has the presently disclosed FFT method has the identical geometry. In this way, the group loop and its associated overhead, have been eliminated. Thus, the disclosed technique eliminates the triple loop nesting structure of the traditional FFT method. As discussed above, the elimination of the control and overhead necessary for exiting and entering the group loop will concomitantly decrease the complexity of the memory address calculations needed to execute the program and thereby increases program efficiency. In addition in the presently disclosed FFT method, the input data 608 is not re-ordered, and the output data values for each stage are properly ordered by the use of the unity stride memory operations discussed above. Thus, the re-ordering that is necessary in the traditional FFT method is eliminated, along with, the additional memory read/write operations. The decrease in the complexity of the calculations needed for the memory read/write operations improves the efficiency of the

operation of the presently disclosed method over the traditional FFT methods as well.

As shown in Figure 6, the first stage 602 uses four twiddle factors 612 all of which equal  $W_8^0$ . The second stage 614 uses one pair of twiddle factors 614,  $W_8^0$  and  $W_8^2$ , twice in a row in that order. The third stage 606 uses four twiddle factors  $W_8^0$ ,  $W_8^2$ ,  $W_8^1$ , and  $W_8^3$  in that order.

Thus, the number of different twiddle factor values used per stage increases as a power of two and the twiddle factor values are retrieved from memory in a bit-reversed order. Therefore, in a preferred embodiment, the twiddle factors are stored in bit-reversed order to simplify the memory operations and increase efficiency. Figure 8 is a schematic illustration of the bit reversal process used to store four twiddle factors for the presently disclosed FFT technique.

In one embodiment, the FFT method is programmed for execution in a general-purpose computer using C, Java, or C++ or other suitable high level language. In another embodiment, the FFT method is programmed for use in a digital signal processing (DSP) system. In the DSP embodiment, the FFT method would use four pointer registers, and preferably the pointer register used to store the twiddle factors in bit reversed order is a circular address register. In addition, an out of place buffer placement of intermediate values is employed to eliminate instability within the inner, butterfly, loop.

As an example, for a 256-point radix-2 FFT, there will be 128 butterfly calculations in 8 stages. Assuming 3 cycles per butterfly, this will require a minimum of  $128*8*3=3072$  cycles. This number is not achievable however,

because of the loop overhead and the overhead in the butterfly setup. Empirical tests have measured a traditional FFT method as using approximately 6600 cycles. The present FFT technique uses approximately 3330 cycles, i.e., nearly a 50% reduction in the number of cycles. Listed below is exemplary simulation code in the C programming language for one embodiment of the presently disclosed FFT method.

```

10  For (int s = 0; s < lgn; s++){           Stage Loop
      For (int k = 0; s < no2; k++){         Butterfly Loop
          R0 = *i0++; r1 = i1++; r3 = i3++;
          real(r4) = real(r1)*real(r3)-imag(r1)*imag(r3);
          imag(r4) = real(r1)*image(r3)+image(r1)*real(r3);
15      real(r5) = real(r0) + real(r4);
          imag(r5) = imag(r0) + imag (r4)
          real(r6) = real(r0) - real(r4);
          imag(r6) = imag(r0) - imag(r4);
          *i2++ = r5;
20      *i2++ = r6;
      }
      p0 <= 1; 13.L = p0;                     Unity Power Update
      i1.B = i2.B; i2.B = i0.B; i0.B = i1.B   Exchange
    }.
25

```

Where lgn is the  $\log_2(N)$ . Accordingly, there is no group loop and the twiddle factors are updated in the unity power update step.

Those of ordinary skill in the art will appreciate that variations to and modifications of the above-described FFT methods and apparatus may be made without departing from the inventive concept disclosed herein. Accordingly, the invention should be viewed as limited solely by the scope and spirit of the appended claims.

35

CLAIMS

1. A method for computing an FFT, the method comprising:
- (a) receiving a plurality of first data values, said  
5 first data values having a total of N-data points;
- (b) storing in a first memory each of said plurality of  
first data values;
- (c) providing in a second memory a plurality of twiddle  
factors stored in sequential locations in a bit reversed  
10 order;
- (d) reading R input butterfly data values of said  
plurality of first data values where each of said R  
butterfly data values are separated by N/R first data values  
in said plurality of first data values;
- 15 (e) performing a radix R butterfly calculation on said  
R butterfly input data;
- (f) providing R butterfly output data values;
- (g) sequentially storing said R butterfly output data  
values in a third memory;
- 20 (h) performing said steps (c) - (g)  $N/R \times 2$  times.
2. The method as in claim 1 further comprising the steps  
of:
- replacing said plurality of first data values in said  
25 first memory with said plurality of data in said second  
memory location;
- repeating steps (c) - (h) a total of  $\log_r(n) \times$  times.
3. The method as in claim 1, wherein  $R=2$ .
- 30 4. The method as in claim 1, wherein said  $R=4$ .

a first memory for storing a plurality of N input data values, said plurality of N input data values being stored sequentially in a time-ordered manner;

a third memory for storing a plurality of output data values; and

20 memory.

7. The apparatus of claim 5 wherein R equals 4.

comprising:

30 points;

said DSP operative to store in a first memory each of  
said plurality of first data values;

said DSP operative to provide in a second memory a  
plurality of twiddle factors stored in sequential locations  
5 in a bit reversed order;

said DSP operative to read R input butterfly data  
values of said plurality of first data values where each of  
said R butterfly data values are separated by N/R data  
points in said plurality of first data values;

10 said DSP operative to perform a radix R butterfly  
calculation on said R butterfly input data;

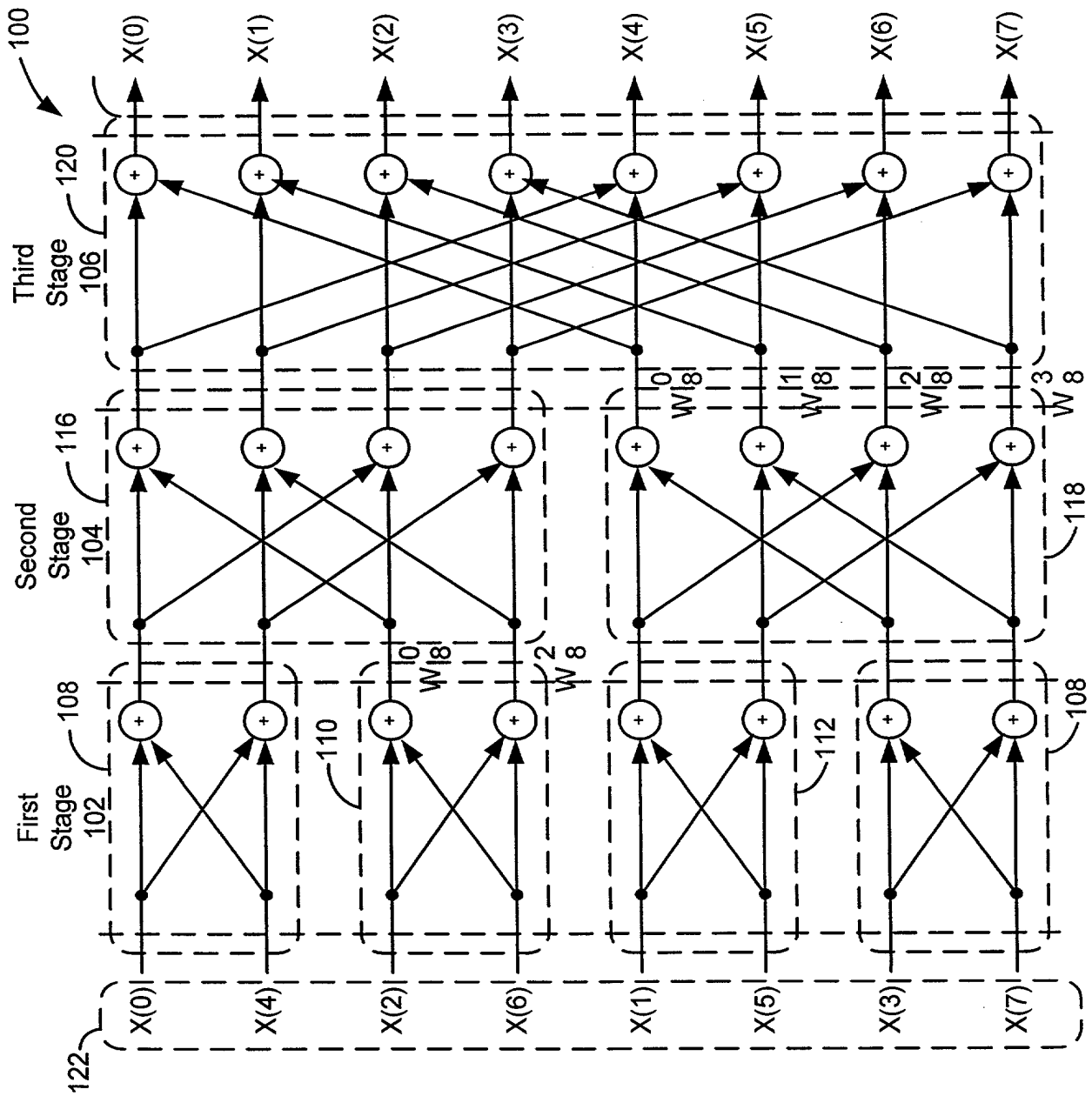
said DSP operative to provide R butterfly output data  
values; and

said DSP operative to sequentially store said R  
15 butterfly output data values in a third memory.

A method for computing an out of place FFT in which each stage of the FFT has an identical signal flow geometry. In each stage of the presently disclosed FFT method the group loop has been eliminated, the twiddle factor data is stored in bit-reversed manner, and the output data values are stored with a unity stride.

ATTORNEY DOCKET NO. ADI-005XX  
WEINGARTEN, SCHURGIN,  
GAGNEBIN & HAYES, LLP  
-TEL. (617) 542-2290

**Fig. 1**  
**Prior Art**





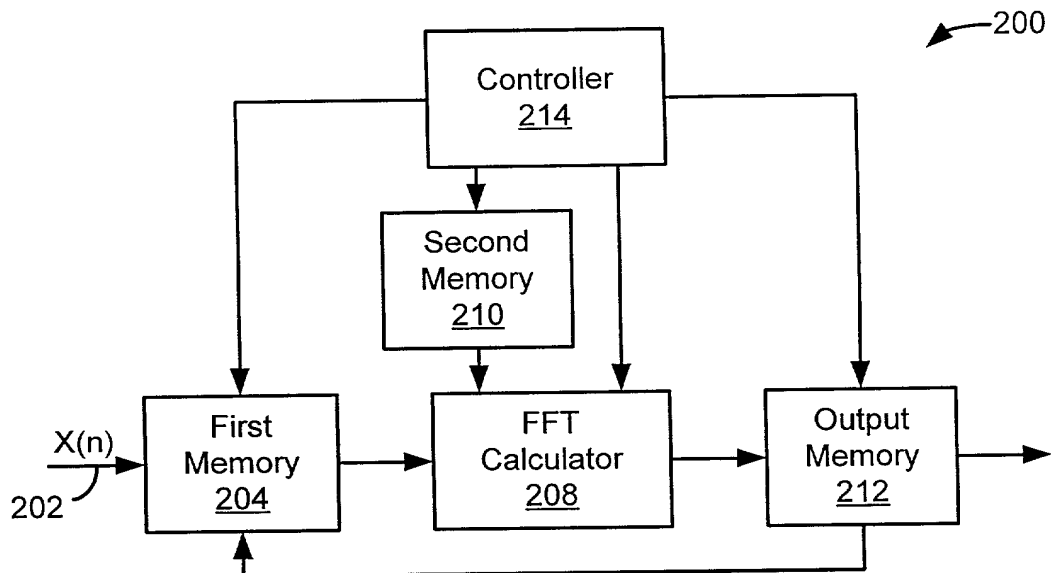


Fig. 2

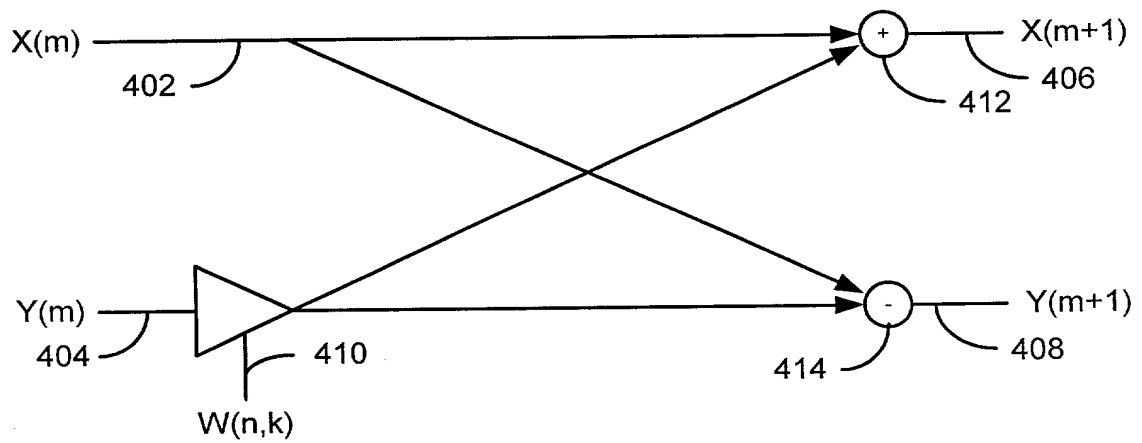


Fig. 4

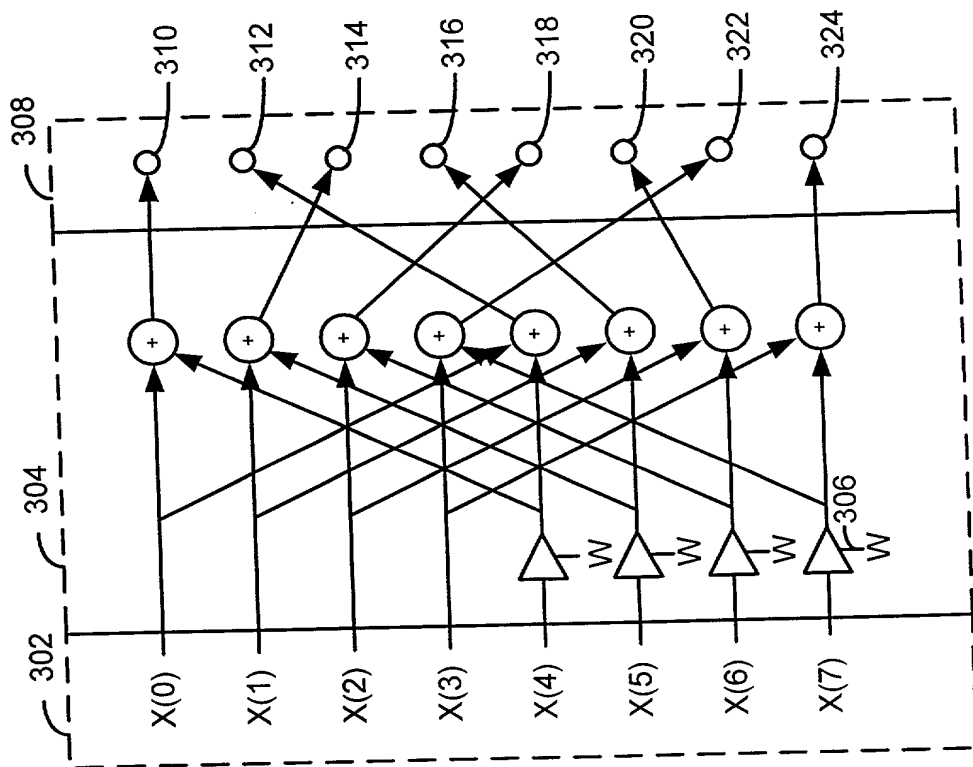


Fig. 3

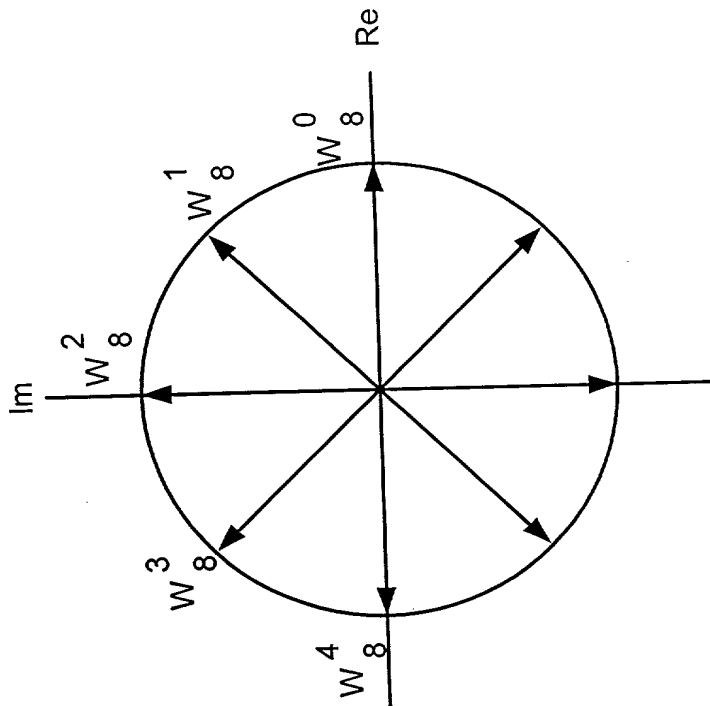


Fig. 5

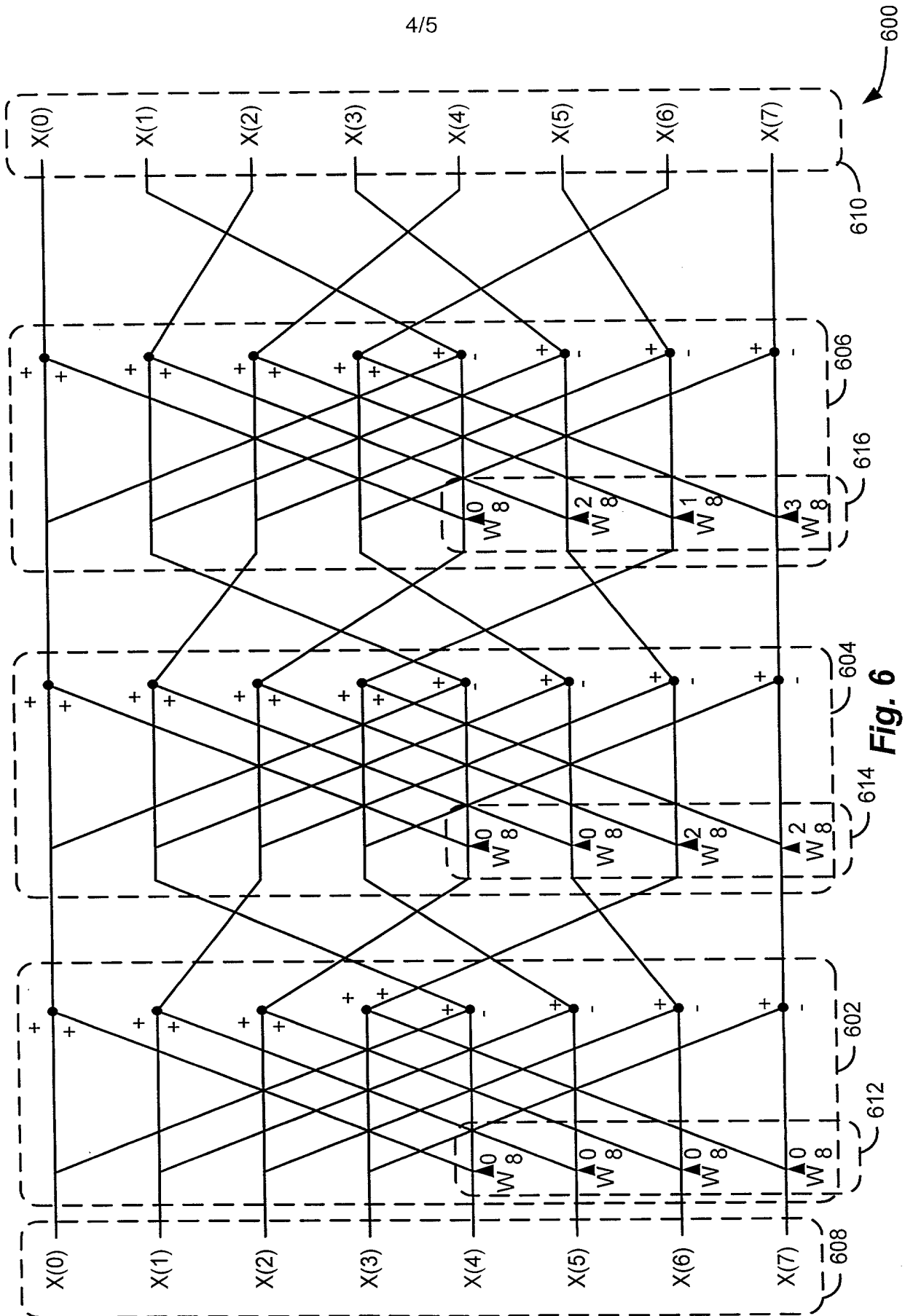


Fig. 6

Input Order	Twiddle Factors	Bit-Reversed Order	Twiddle Factors
0 0 0	$W_8^0$	0 0 0	$W_8^0$
0 0 1	$W_8^1$	1 0 0	$W_8^4$
0 1 0	$W_8^2$	0 1 0	$W_8^2$
0 1 1	$W_8^3$	1 1 0	$W_8^6$
1 0 0	$W_8^4$	0 0 1	$W_8^1$
1 0 1	$W_8^5$	1 0 1	$W_8^5$
1 1 0	$W_8^6$	0 1 1	$W_8^3$
1 1 1	$W_8^7$	1 1 1	$W_8^7$

**Fig. 7**

**DECLARATION AND POWER OF ATTORNEY**

As a below-named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled: A METHOD FOR EFFICIENTLY COMPUTING A FAST FOURIER TRANSFORM

the specification of which (check one):

☒ is attached hereto. ☐ was filed \_\_\_\_\_ as Application No. \_\_\_\_\_  
amended on \_\_\_\_\_ (if applicable).

☐ was filed as PCT International Application No. \_\_\_\_\_ on \_\_\_\_\_,  
and was amended under PCT Article 19 on \_\_\_\_\_ (if applicable).

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations §1.56(a).

I hereby claim foreign priority benefits under Title 35, USC §119(a)-(d) of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

<u>Prior Foreign Application(s)</u>	<u>Date Filed</u>	<u>Priority Claimed</u>	
_____ (Number) (Country)	_____ (Day/Month/Year)	<input type="checkbox"/> Yes	<input type="checkbox"/> No
_____ (Number) (Country)	_____ (Day/Month/Year)	<input type="checkbox"/> Yes	<input type="checkbox"/> No

I hereby claim the benefit under Title 35, USC §119(e) of any United States provisional application(s) listed below:

_____ (Application Number)	_____ (Filing Date)
_____ (Application Number)	_____ (Filing Date)
_____ (Application Number)	_____ (Filing Date)

Express Mail Number

EL41842505005

Attorney

Docket No.: ADI-005XX

I hereby claim the benefit under Title 35 USC §120 of any United States application(s) listed below and insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35 USC §112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Application No.)	(Filing Date)	(Patented/pending/abandoned)
-------------------	---------------	------------------------------

(Application No.)	(Filing Date)	(Patented/pending/abandoned)
-------------------	---------------	------------------------------

**POWER OF ATTORNEY:** As a named inventor, I hereby appoint the following attorney(s) to prosecute this application and transact all business connected therewith in the Patent and Trademark Office, and to file with the USRO any International Application based thereon.

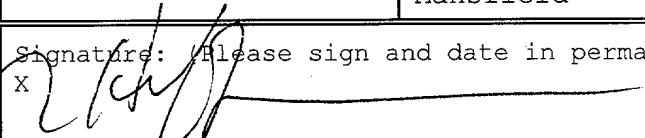
Stanley M. Schurgin, Reg. No. 20,979  
 Charles L. Gagnebin III, Reg. No. 25,467  
 Paul J. Hayes, Reg. No. 28,307  
 Victor B. Lebovici, Reg. No. 30,864

Eugene A. Feher, Reg. No. 33,171  
 Beverly E. Hjorth, Reg. No. 32,033  
 Holliday C. Heine, Reg. No. 34,346  
 Gordon R. Moriarty, Reg. No. 38,973

Address all correspondence to:

**WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP**  
 Ten Post Office Square  
 Boston, Massachusetts 02109  
 Telephone: (617) 542-2290  
 Telecopier: (617) 451-0313

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of First Inventor: Marc Hoffman		
City of Residence Mansfield	State or Country Massachusetts	Country of Citizenship United States
Post Office Address 99 Pleasant Street	City Mansfield	State or Country Zip Code Massachusetts 02048
Signature: (Please sign and date in permanent ink.) X 		Date signed: X 7/26/2000

